AFRL-IF-RS-TR-2006-326
Final Technical Report
November 2006

KOJAK: SCALABLE SEMANTIC LINK DISCOVERY VIA INTEGRATED KNOWLEDGE-BASED AND STATISTICAL REASONING

The University of Southern California

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory Rome Research Site Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-IF-RS-TR-2006-326 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/	/s/
DEBORAH A. CERINO Work Unit Manager	JOSEPH CAMERA Chief, Information & Intelligence Exploitation Division Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
NOV 06	Final	Aug 01 – Mar 06

4. TITLE AND SUBTITLE
KOJAK: SCALABLE SEMANTIC LINK DISCOVERY VIA INTEGRATED KNOWLEDGE-BASED AND STATISTICAL REASONING

5a. CONTRACT NUMBER
F30602-01-2-0583

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
31011G

6. AUTHOR(S)
Hans Chalupsky

5d. PROJECT NUMBER
EELD

5e. TASK NUMBER
01

5f. WORK UNIT NUMBER
04

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Southern California/Information Sciences Institute
4676 Admiralty Way
Marina Del Rey California 90292

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/IFED
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2006-326

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA #06-748

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Link discovery (LD) is a new challenge in data mining whose primary concern is to identify strong links and discover hidden relationships among entities and organizations based on low-level, incomplete and noisy evidence data. Within this effort, USC/ISI addressed this challenge by developing a hybrid link discovery system called KOJAK that combines state-of-the-art knowledge representation and reasoning (KR&R) technology with statistical clustering and analysis techniques from the area of data mining.

15. SUBJECT TERMS
Knowledge representation and reasoning, statistical clustering and analysis

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Deborah A. Cerino
U	U	U	UL	53	19b. TELEPHONE NUMBER (*Include area code*)

Table of Contents

Table of Figures

List of Tables

1 Introduction

Link discovery (LD) is a new challenge in data mining whose primary concern is to identify strong links and discover hidden relationships among entities and organizations based on low-level, incomplete and noisy evidence data. Within this program we addressed this challenge by developing a hybrid link discovery system called KOJAK that combines state-of-the-art knowledge representation and reasoning (KR&R) technology with statistical clustering and analysis techniques from the area of data mining. Using KR&R technology allows us to represent extracted evidence at very high fidelity, build and utilize high quality and reusable ontologies and domain theories, have a natural means to represent abstraction and meta-knowledge such as the interestingness of certain relations, and leverage sophisticated reasoning algorithms to uncover implicit semantic connections. Using data or knowledge mining technology allows us to uncover hidden relationships not explicitly represented in the data or findable by logical inference, use clustering techniques to find sets of entities that are temporally or organizationally related, or use clustering simply to improve efficiency by carving up a very large data space into smaller more manageable pieces.

KOJAK is built on top of PowerLoom™, a knowledge representation system that provides a language and environment for constructing intelligent applications. PowerLoom™ uses a fully expressive, logic-based representation language, and it uses a natural-deduction-style backward and forward chainer as its inference engine. PowerLoom is written in STELLA, a new programming language developed by our group that can be translated into Lisp, C++ and Java. KOJAK consists of three major components:

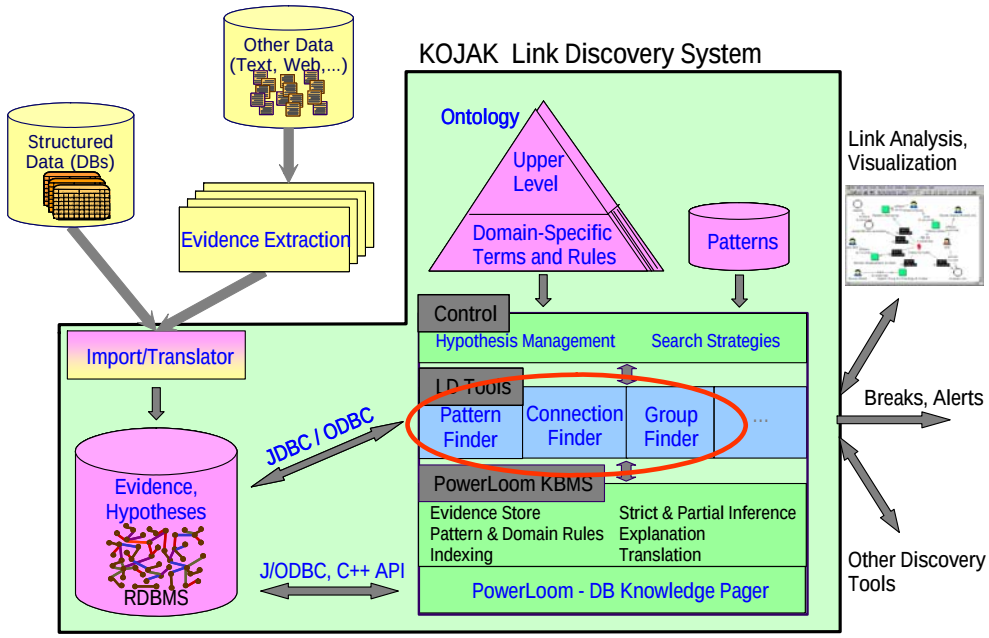


Figure 1: KOJAK Architecture

1 **KOJAK Pattern Finder**, which detects patterns and events

2 **KOJAK Connection Finder** which finds abnormal or "interesting" entities and connections

3 **KOJAK Group Finder** which detects and extends groups of agents with similar behavior or strong associations (based on mutual information, connectivity, etc.)

The architecture of the system is shown in Figure 1. In the following we describe each module in a bit more detail.

1.1 Pattern Finder

Using a full-fledged KR&R system allows us to easily represent and reason with different levels of abstraction, which is of primary value in the area of link discovery. For example, we can infer an *is-associated-with* relation from more specialized relations such as *is-employee-of, is-member-of, is-leader-of*, etc., and we can also specify patterns of interest in terms of these more abstract and higher-level relations. It also affords us other

```
(FORALL (?ev ?c ?v ?m ?h)
  (=> (AND (MurderForHire ?ev)
           (victimIntended ?ev ?v)
           (hitContractor ?ev ?c)
           (mediators ?ev ?m)
           (hitman ?ev ?h))
      (contractKill ?ev ?c ?v ?m ?h)))
```

Figure 2: Example of pattern rule

conveniences such as a natural way of representing patterns or scenarios of interest, powerful inference mechanisms, e.g., to discover implicit semantic relationships or to pinpoint missing information, as well as methods for hypothetical reasoning to evaluate potential relationships. The KOJAK Pattern Finder matches fragmented evidence pieces against given patterns and evaluate and score matches to minimize false positives and false negatives. The Pattern Finder is built on top of our PowerLoom KR&R system and uses high-fidelity logic-based representation of evidence and complex patterns in addition to hypothesis generation, testing and explanation via partial logical inference (abduction).

1.2 Connection Finder

A significant portion of knowledge discovery and data mining research focuses on finding patterns of interest in data. Once a pattern is found, it can be used to recognize satisfying instances. The new area of link discovery requires a complementary approach, since patterns of interest might not yet be known or might have too few examples to be learnable. To address this problem we developed the KOJAK Connection Finder (aka "UNICORN") which is an unsupervised link discovery method aimed at detecting interesting nodes or interestingly-connected nodes in multi-relational datasets. Interestingness is modeled via abnormality of "semantic profiles" that are based on how often similar paths occur in the data.

Our experiments show that our program can find interesting connections in a network without having to learn the patterns of interestingness beforehand. The key advantage of our method over the state-of-the-art is that it does everything in an unsupervised manner and eliminates the necessity to regenerate new rules or new training data for different queries or even when the whole domain is changed. It also eliminates the risk of being biased by the apparent meaning of link types. Another advantage of our approach is that it can focus the user's attention on events that are otherwise hard to notice. The inspirations triggered by such alerts can sometimes lead to the discovery of patterns or other knowledge. The final advantage to our approach is that it is a general-purpose method and can be applied to arbitrary multi-relational datasets. The Connection Finder

was one of the early technology nuggets in the program and was awarded second place in the Open Task of the 2003 KDD Cup (Lin & Chalupsky, 2003).

1.3 Group Finder

The KOJAK Group Finder (GF) is a hybrid logic-based/statistical LD component designed to solve group detection problems (Adibi et al., 2004; Adibi & Chalupsky, 2005). The system takes primary and secondary evidence as input and produces group hypotheses with ranked lists of group members as output. Primary evidence is usually lower volume, high reliability and owned by the intelligence organization, while secondary evidence (e.g., news articles on theWeb, etc.) is not owned, can be very large scale, and is usually subject to querying restrictions.

These scale and access restrictions heavily influenced the architecture of the Group Finder which works in four phases. First, a logic-based group seed generator analyzes the primary evidence and outputs a set of seed groups using deductive and abductive reasoning over a set of domain patterns and constraints. Second, an information-theoretic mutual information model finds likely new candidates for each group, producing an extended group. It does so by looking for people that are strongly connected with one or more of the seed members. Computing connection strength is achieved by a mutual information model which exploits data or evidence such as individuals sharing the same property (e.g., having the same address) or being involved in the same activity (e.g., sending email), etc. Third, the mutual information model is used to rank these likely members by how strongly connected they are to the seed members. Fourth, the ranked extended group is pruned using a threshold to produce the final output. Table 1

	Y2	Y2.5	Y3
Entities	10,000	10,000	100,000
Links	100,000	100,000	1,000,000
F-value	0.505	0.680	0.498
Performance	1st Place	1st Place	1st Place

Table 1: Group Finder Performance

summarizes performance results of the Group Finder over the most recent three program-wide evaluations.

1.4 Potential Applications

KOJAK's link discovery tools are applicable in a variety of situations. First and foremost, they should help intelligence analysts to do their job by finding groups of potential terrorists or suspect individuals with abnormal and unusual characteristics that might warrant further investigations. Other areas of application are, for example, container security to locate and discover suspicious containers having information about countries, ports, content, route and other relevant information, and fraud detection, for example, to detect complex suspicious activities based on known patterns of fraudulent activity.

2 KOJAK Connection Finder (aka UNICORN)

One challenging problem in the area of link discovery (and intelligence analysis) is to find things of interest without knowing exactly what one is looking for. While sometimes patterns are available to guide the search of an analyst or link discovery tool, these patterns will only lead to instances of known activity but will not pick up never before seen patterns of suspicious activity. To address this problem, we built the KOJAK Connection Finder (now called UNICORN), which is a discovery tool that performs "Interesting Instance Discovery" in multi-relational datasets (or semantic graphs).

In the following we describe UNICORN[1], which is an unsupervised instance discovery framework that finds interesting instances in multi-relational datasets by identifying those nodes with an abnormal *semantic profile*. UNICORN is able to transform an instance discovery problem in a semantic network into a numerical outlier detection problem by summarizing the semantic graph structure surrounding a particular instance. The experiments performed on a real-world bibliography dataset show that it is indeed able to find instances that are interesting in real life in a completely unsupervised manner. Potential application areas for our framework are inspiration-driven discovery, homeland security, law enforcement, and data cleaning.

2.1 Motivation

Machine discovery has been an important research area of AI for more than twenty years. Herbert Simon described it as "gradual problem-solving processes of searching large problem spaces for incompletely defined goal objects" (Simon, 1995). The majority of machine discovery programs focus on discovering (or rediscovering) the theories and laws of natural science which can be viewed as search for parsimonious description of the world (Milosavljevic, 1995). Most science discovery programs rely on some pre-requisite

[1] UNICORN is an abbreviation for "UNsupervised Interesting-instance disCOvery in multi-Relational Network".

knowledge in a specific domain and some general knowledge or heuristics to guide search.

More recently, researchers encountered a new problem: there is more and more data produced and stored that we do not know how to analyze or interpret. Thus a new type of discovery research emerged called knowledge discovery and data mining (KDD). KDD focuses on discovering and extracting previously unknown, valid, novel, potentially useful and understandable patterns from lower-level data (Fayyad et al., 1996).

The major difference between machine discovery in science and KDD is that the former is mainly knowledge driven and the later is data driven. In science discovery, the key challenge lies in how to model a specific domain as well as how to encode appropriate knowledge (heuristics) to guide the search, while in KDD the main focus is on extracting useful information (patterns) from large, sometimes heterogeneous and usually noisy data sets.

To address the challenges of link discovery we investigate a new type of discovery problem that lies somewhere between machine discovery for science and KDD. We call it the "interesting instance discovery (IID)" problem, which focuses on discovering interesting or abnormal instances in large multi-relational datasets. While interesting instance discovery can be viewed as a form of data mining, it is different from traditional KDD since it does not focus on finding regularities or patterns in the data. It also does not aim to discover scientific laws as in automated science discovery.

The main challenge of IID arises from the fact that the term "interestingness" is vague and there is no consensus on how it can be measured. Therefore training on "interesting" examples is not a suitable approach for IID problems: if there is a systematic way for us to find **unbiased** examples for training, we can simply implement it to be our IID tool. Moreover, training on biased examples will only generate a system that produces results "similar" to the biased examples. In this sense the system is more like a learning system that learns the example generator's bias instead of a discovery system that finds the truly

useful interesting results. Nevertheless, the incompletely defined goal object (i.e. interestingness) makes IID a discovery problem instead of an easier learning problem, and consequently prohibits us from using any supervised learning methods. In addition, the lack of universally accepted interestingness measures also creates a very difficult problem on how to verify the results produced by the IID tools.

In a nutshell, discovering interesting instances addresses the problem of finding interesting things without knowing exactly what one is looking for. The focus of our work is on discovering interesting instances in large multi-relational dataset without utilizing any training examples. A multi-relational dataset can be represented as a semantic network such as the one shown in Figure 3. The network consists of a set of nodes representing objects and links representing semantic relationships between them. Also our framework is designed for large network not only in terms on the number of entities and links but in terms of the various types of relationships between the entities.

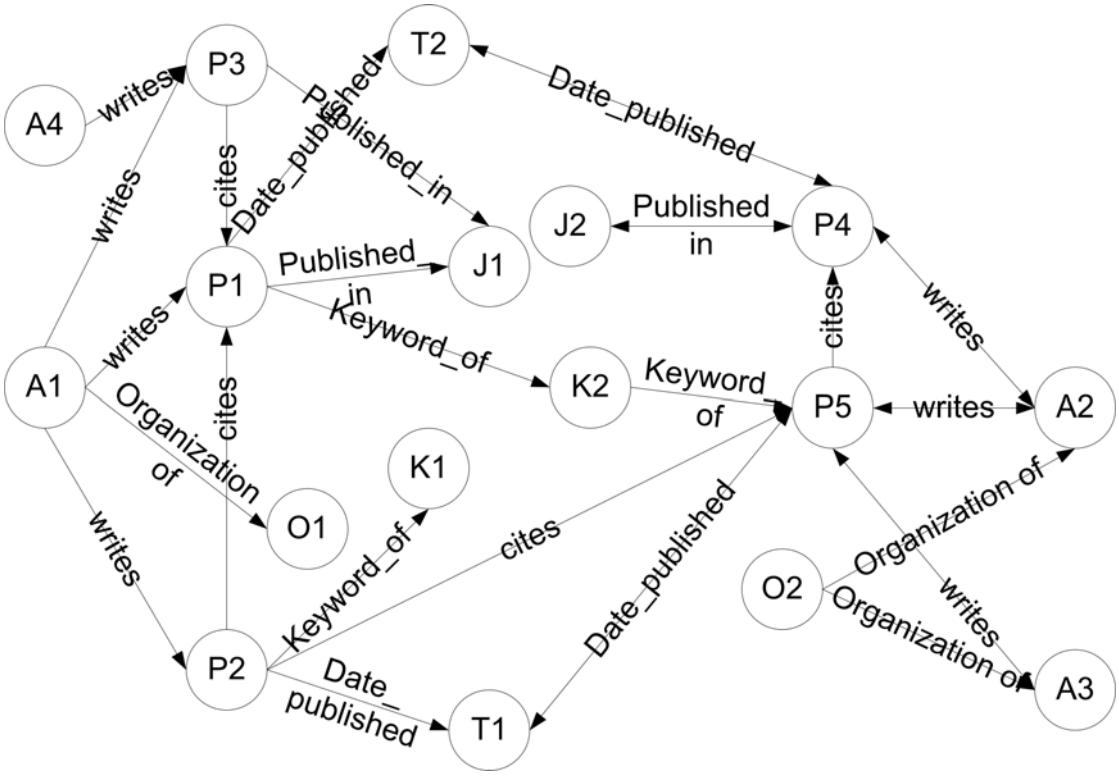


Figure 3: A semantic network in a bibliography domain

There are three major characteristics that distinguish IID research from the typical KDD and science discovery research. First it utilizes **unsupervised** methods so no training example is needed, second it discovers **interesting instances** instead of patterns, and third it is applicable to **multi-relational** datasets instead of numerical datasets.

2.2 Interesting Instance Discovery

What generally makes an instance interesting? Our main intuition is that instances are interesting to a human observer if they are **abnormal**. There are two issues with this statement that we need to address: First, what does it mean for an instance to be "abnormal" and why are abnormal instances often interesting? Say if there is a "property" (e.g. the salary) to characterize the instances, then the abnormal instances are those that have unusual values for that property (e.g. billionaires). To be more general, given there are a set of "features" to characterize an instance, the abnormal instances are those have

different “combinations” of feature values. For example, if people are portrayed by their behavior, then the billionaire who gives up all his money might be abnormal. Note that abnormal is not necessary identical to rare. For example, in the sequence (1, 100, 100, 100, 100, 100, 1000, 10000), the abnormal number might be 100, since it occurs more frequently than any of the others, while the rare ones are those that occur only once. It is hard to “prove” that abnormality represents interestingness. But empirically we have observed that (as will be elaborated in the discussion of experiments) those logically inconsistent or not-so-plausible behaviors (e.g. billionaires throwing away all their money) in general catch an investigator’s eye and imply interestingness. Note that we also do not claim that all the abnormal instances are inconsistent or implausible ones, however, we can say that if there are such instances (which conceivably are interesting), then the chance will be great that they can be discovered by looking for the abnormal ones.

Secondly, “interestingness” in many cases is domain dependent. Namely one instance can be interesting in one domain but not in another. For example, the evidence “X published a paper with Y” might be more interesting if it appears in a police murder dataset compared with a bibliography dataset. Using abnormality to model interestingness in an unsupervised manner in fact can take the context dependency into account since it extracts the abnormal instances by comparing it with the others in the same context.

2.3 Finding Interesting Instances in Semantic Graphs

Below we describe how the UNICORN framework can discover interesting instances in a multi-relational dataset without relying on background knowledge or training examples. Using this approach it can answer two general types of questions:

1. Given a multi-relational dataset such as the graphic representation of the bibliography network shown in Figure 3, which are the *k* most interesting instances (nodes)? For example, identify the most interesting author in the network.

2. Given a network and a specific source node, find the *k* most interesting nodes connected to it. For example: find the organization an author A1 is most interestingly connected to.

The general idea is that in a semantic graph a node or instance is abnormal if it has significantly different meaning or *semantic profile* than any of the other nodes. There are two challenges for this approach:

1. How can we capture the semantic profile of an instance (i.e. the meaning or information it carries) without knowing the semantics of the particular nodes and links?
2. If we somehow can represent the semantic profile of instances, how can we quantify the semantic differences between nodes in order to determine the abnormal ones?

To address the first question we start with the following observation: each path in the network can be translated into standard logical notation by representing nodes as constants and links via binary predicates. For example, in Figure 3, the path "P2 cites a paper P1 that is published in J1" can be represented as cites(P2,P1) $\wedge$ published_in(P1,J1). This logical expression partly characterizes the meaning of the nodes P1, P2 and J1. It only partly characterizes it, since there are many other paths (or logical expressions) that also involve these nodes. In our view, it is the *combination of all paths* or expressions a node participates in that define the meaning or semantic profile of a node. This is different from standard treatments in logic where the semantics of a constant (or node) is simply taken to be its denotation but more similar to treatments in the semantic network literature where the semantics of a node is viewed to be determined by the whole network it is in (e.g., see (Hill, 1995)).

Given this view we can now model the semantic profile of a node (and link) by treating all paths it participates in as binary features. For example, assuming a network such as the one in Figure 3 contains a total of 100 different paths, then each node (and link) can be represented by a 100-dimensional feature vector. With such a representation, we can

then use standard vector space similarity or outlier-detection algorithms to look for abnormal nodes.

While the previous paragraph describes the central idea underlying the approach used for UNICORN, there are some issues with it that still need to be addressed. The first issue is that different paths should not necessarily be viewed as distinct. It generates an overfitting problem. Since each path is unique, the only nodes sharing a particular path feature would be those participating in the path, which would make this type of features useless to associate the nodes inside the path with the ones outside of it. For example, the two paths cites(P2,P1) $\wedge$ published_in(P1,J1) and cites(P2,P1) $\wedge$ published_in(P1,J2) might be important to compare and contrast J1 and J2, however, since they would become independent features they could not really contribute to a meaningful comparison.

The second issue relates to complexity: a large semantic network can easily contain millions of paths, and computation in such a very high dimensional space could be costly. The third issue has to do with explanation: ideally, a human analyst would like to get an answer why the discovery tool picked a certain instance as interesting or abnormal as we did in our experiments. However, providing such an explanation from a very high-dimensional feature set is difficult.

These issues motivate the search for a more condensed feature set without losing the ability to capture the major meaning profile of instances. We do this by defining equivalence classes between different paths that we call *path types* and then use these path types as features instead of individual paths. Whether two individual paths are considered to be of the same path type will depend on one of several similarity measures we can choose. For example, we can view a set of paths as equivalent (or similar, or of the same type) if they use the same sequence of relations. This view would consider the following three paths as equivalent:

cites(P2,P1) $\wedge$ published_in(P1,J1)
cites(P2,P1) $\wedge$ published_in(P1,J2)

cites(P2,P3) ∧ published_in(P3,J1)

Alternatively one can consider a set of paths that go through the same sequence of nodes as equivalent. In this case these two paths “write_letter_to (A, P1)” and “calls (A, P1)” are equivalent.

The next question then becomes how we can generate a meaningful and representative set of path types? Take the path cites(P2,P1) ∧ published_in(P1,J1) for example. There are five ground elements in this path: cites, P1, P2, published_in and J1. If we relax one of its elements, say J1, to a variable X, then we get a new meaning frame cites(P2,P1) ∩ published_in(P1,X) which now represents a more general concept: “paper P2 cites paper P1 that is published in some journal”. Additionally, we could also generalize a link such as published_in which would give us cites(P2,P1) ∧ *y*(P1,X) or “paper P2 cites paper P1 that has something to do with some journal”. In fact we can generalize any combination of nodes or links in a path to arrive at a more general path type. These path types still convey meanings but at a more abstract level which makes them more useful as features to compare or contrast different instances or nodes.

The path type features can be taken as the aggregated version of path features in the sense that each path type contains multiple realizations in the dataset. Take the path type writes(X, Y) as an example: One instance a1 might occur in many paths of this type (say writes(a1, *y1*)…writes(a1,y99), which implies a1 writes 99 papers), while another instance a2 might occur only in a few (say 1 time). Assuming that in the whole dataset only a1 and a2 write papers, then a1 contributes 99%, a2 1% and the rest 0% to this “writing paper” behavior in a given world. We therefore define the *contribution* of an instance *x* to a path type *pt* as the total number of times *x* occurs in paths of type *pt* divided by the total amount of times *pt* occurs in the dataset. The contribution of an instance to a path type is indeed the average occurrence rate of that instance in that path type.

Our final observation is that *abnormal contribution* in many cases captures the idea of interestingness. For example, if most of the people in a dataset write 1 paper per year, then the person who writes 99 papers per year will have a higher chance to become an interesting instance. Thus, to model an instance in the semantic network we treat path types (not paths) as features and their **contributions** to the instance as feature values (instead of using binary features). In other words, the semantic profiles of instances are represented by numeric feature vectors that model the contribution of each path type the instance participates in.

We can now describe how UNICORN solves the first problem of finding the top interesting nodes in a semantic net by ranking them according to their interestingness. It involves these steps:

1. Choose a set of *n* path types to represent the instances (we will describe two ways of automatically choosing this set in the experiment section based on the idea of variable relaxation described above).
2. For each instance, compute the contribution for each chosen path type as the instance's feature value for the particular path type. Repeat this step for all instances in the dataset.
3. Given there are *m* instances, we now have *m* *n*-dimensional points to represent the meaning profile of each instance. Then we can use a standard outlier detection approach to quantify and rank the abnormality or interestingness of each instance (in the experiment we used Ramaswamy's *k*-th nearest distance-based algorithm (Ramaswamy et al., 2000)).

The procedure for solving the second problem of finding the top interesting instances connected to a given source *s* is almost identical. The only difference is that the chosen path types need to reference the source *s* (in other words in step 1 the chosen path type should have *s* grounded somewhere) since it is reasonable to use **only** the meaning frames (path types) that have something to do with *s* to evaluate other instances connected to *s*.

2.4 Experiments

In general we face a chicken and egg dilemma when trying to verify discovery results. For example, if there are unbiased criteria to judge whether the abnormal instances generated by UNICORN are interesting or not, we can simply implement those criteria as our interesting instance finder. The reason for promoting IID research is because there are no such criteria known so far. This chicken and egg dilemma makes it very hard to justify a discovery system.

Nevertheless, there are in general two different paths we can choose to evaluate our discovery system. The first is to create an artificial dataset and manually add some instances that "we" think are interesting in order to test if UNICORN can find them. The second is to apply UNICORN to a real world dataset and try to analyze whether the output of UNICORN is truly interesting or not. We chose the second path to evaluate UNICORN because of several reasons: First of all, since we have claimed one of the advantages of UNICORN is that it cannot be biased by the apparent semantic meanings of the relationships and it can produce unexpected or non-plausible results. However, testing it on an artificial dataset with biased results (it would be biased, because we generated the interesting instances manually) does not reflects and justify its strength. The second reason is that we claim that UNICORN handles large and complex datasets. But generating a large and complex dataset that contains a set of interesting instances is essentially a dual problem to IID. Hence, we believe the first path is more suitable for a supervised learning system and the fair test for UNICORN is to put it in a real world large multi-relational environment and then analyze its results. The drawback of using real world large dataset is that there is no gold-standard solution available. Nobody knows what and how many instances should be discovered as interesting from a dataset that has thousands of nodes and hundred thousands of links.

In our experiments we used a real-world bibliography dataset as the case study. Our first goal is to demonstrate how the UNICORN framework can be applied to a real-world large multi-relational dataset. The second goal is to show that the abnormal instances we found does to some extent capture the meaning of "interestingness".

2.4.1 Experimental Setup

We used the "High Energy Physics - Theory" (HEP-Th) bibliography dataset provided for the 2003 KDD Cup. The data was translated into a multi-relational network as follows: We extracted six different types of nodes (entities) and six types of links (relations) from the dataset to generate the network. Nodes represent paper IDs (29014), author names (12755), journal names (267), organization names (963), keywords (40) and the publication time encoded as year/season pairs (60). Numbers in parentheses indicate the number of different entities for each type in the dataset. We defined the following types of relationships to connect various types of nodes: writes(*a*,*p*), date_published(*p*,*d*), organization_of(*a*,*o*), published_in(*p*,*j*), cites(*p*, *r*), keyword_of(*p*,*k*), where 'a' stands for author, 'p' as paper, 'd' as date, 'o' as organization, 'j' as journal and 'k' as keyword. These links are viewed to be directional with an implicit inverse link. Thus, there are a total of 12 different relations. The network generated is similar to the one in Figure 3, only that there are 43095 different nodes and 477423 links overall.

We choose Ramaswamy's algorithm for outlier detection. This algorithm ranks the outlier points by their Euclidean distance to the *k*-th nearest neighborhood. That is, the outliers are those far away from their *k*-th neighbors (it is allowable to have *k* points around an outlier point). In our experiments we use two different ways to evaluate the discovered results: (1) We first examine the original network to learn the reason why instances are chosen as outliers. UNICORN does not have any knowledge about the semantics of the nodes, but manually inspecting which path types contributed how much together with our knowledge of what the meaning of path types is a good way of evaluating whether the program has indeed found something interesting. (2) We use the Web as an external source to find supporting evidence. Since the nodes represent real-world entities such as people, we can "verify" the computed results by investigating whether they reflected a real-world, semantically interesting profile or connection visible through the World-Wide Web.

2.4.2 Finding Interesting Instances

The goal for this experiment is to find interesting instances (e.g. interesting people) in the bibliography dataset by abnormality analysis. The first step is to characterize the path types to choose as features to represent instances. For this experiment we chose loop paths of length at most five that lead from an instance back to itself. We consider two loop paths of the same type if they go through the same sequence of relations. The set of all such different loop path types in the data constitutes the total list of features to consider and can be determined automatically by UNICORN by scanning the data for which loop types actually occur. The reason we added the additional loop constraint for this experiment is that loop paths say more about a particular instance, since it is mentioned twice. For example, the path writes(X,P2) $\wedge$ cites (P2,P3) $\wedge$ writes(X,P3) paraphrased as "X writes a paper P2 that cites his other paper P3" says more about X than the path writes(X,P2) $\wedge$ cites (P2,P3) $\wedge$ writes(Y,P3) paraphrased as "X writes a paper P2 that cites paper P3 written by Y". Note that according to our path type interpretation, the path writes(X,P4) $\wedge$ cites (P4,P5) $\wedge$ writes(X,P5) has the same path type as the loop described above. We also restricted the loop length to be at most five to (1) bound the number of features to consider, but also (2) since longer and longer paths represent more and more arbitrary and literally far-fetched semantic relationships.

The query to be asked is who among the 12755 authors in this dataset are the most interesting ones? The ranking generated by UNICORN shows C.N. Pope, Ashoke Sen, and Edward Witten at the top of the list. After looking into the data and feature distribution, we find that the reason why C.N. Pope is chosen is twofold: First, he contributed significantly in most of the loop types. However this fact itself is not enough to distinguish him from other nodes that also contribute significantly. The second reason is that he contributes 0 to the loop: organization_of(*x,o1*) $\wedge$ organization_of(*y,o1*) $\wedge$ organization_of (*y,o2*) $\wedge$ organization_of(*x,o2*). That is, UNICORN finds that there is no other person in the data that has ever belonged to any two organizations he has ever worked in which is abnormal for people who contribute significantly in most other dimensions. Ashoke Sen is chosen as abnormal because some loops suggest he has very focused research directions (e.g. he contributes the most to the loop "a single paper cites

multiple of his papers") while some suggests he has a broad research directions (e.g. he contributes relatively low to the loop "his papers are published in the same journal") which is not common at all in this data. The reason Edward Witten is chosen, in short, is because he did not contribute much for most loop types (e.g. he does not publish or co-author as frequently as others in this data set), but also that most people in this data cite more than one of his publications. After searching on the web we found that Edward Witten is a famous mathematical physicist who has won the Fields Medal, the highest honor a mathematician can receive. This fact strengthens the validity of our discovery, since even though his research is not fully focused on high-energy physics, some of his contributions to the fundamental mathematics must be valuable to this community and thus attract many citations.

2.4.3 Finding Interestingly Connected Instances

In this experiment, we tried to answer the query which nodes are interestingly connected to a given source S, where interestingness is modeled as abnormality. Abnormality is defined based on the contribution of various path types from an instance **connected** to the source. Again, the first step is to characterize the path types to choose as features to characterize instances. For this experiment we choose paths of length at most four and we consider two paths to have the same type if they follow the same sequence of relations. For example, the following two paths are of the same path type under this interpretation: "cites(S,P1) $\wedge$ published_in(P1,J1)" and "cites(S,P5) $\wedge$ published_in(P6,J1)".

This choice is plausible, since in general relations carry more information than entities unless some of the entities have special meaning to an analyst. By using these ordered relations as path types, we are able to reduce the number of features from thousands to less than 100. As before, under this constraint the total set of features to consider can be determined automatically by UNICORN by analyzing which path types actually occur for particular nodes.

We started by picking C.N. Pope as the source node, since in this dataset he is the one with the most publications, which provides us with a rich number of connections to other nodes. The first query we chose was **"which organizations are interestingly connected with Mr. Pope?".** The results show that U. Texas A&M is the most interesting one, followed by SISSA and the third INFN. After analyzing the data we found that the major reason UNICORN regards U. Texas and SISSA to be the outliers is that among the 963 organizations, Pope uses email addresses from only these two institutions. Both institutions contribute 50% in this direction while others contribute 0, which makes them special. However, the reason it considers INFN as an outlier is different. It is due to the combination of two pieces of evidence:

1. In the majority of institutes, the two path types "Pope's colleagues have ever belonged to that institute" and "Pope's co-author belongs to that institute" are positively correlated with respect to the contribution (that also implies that Pope writes many papers with his colleagues).
2. Although the institution INFN has the highest contribution (8.5%) in the first path type shown above, it has 0% contribution as to the second one (it has no members co-authoring with Pope).

Combing these two facts, our program discovered that INFN is different from others to Pope. In other words, INFN is chosen because Pope tends to write papers with his colleagues but he has never written any with his colleagues that have ever belonged to INFN, despite the fact that INFN produces the most people belonging also to Pope's institution (8.5%). Intuitively this is an interesting and unexpected discovery and might potentially trigger new findings (e.g. "People from INFN focus on slightly different research topics than Pope", or "Pope does not like People from that institute").

After investigating through the Web by combining two interestingly connected nodes as search keywords (e.g. "C.N.Pope SISSA"), we found that Dr. Pope is a professor at U. Texas A&M. He probably was at SISSA, Italy during Fall 1994 and Summer 1996, since his email and mailing address were changed to SISSA during that period.

Next we tried to see how our program performs when using another person as the source. We randomly selected a person, Dr. Chiang-Mei Chen, who has a smaller amount of HEP-Th publications (20) and asked UNICORN the query **"Which organizations are interestingly connected to the person Chiang-Mei Chen?"** The results signify that the school NCU (National Central University, Taiwan) is the 1st outlier followed by MSU (Moscow State University) and NTU (National Taiwan University). After analyzing the HEP-Th data, we found that they are the only organizations that Dr. Chen has ever belonged to. However, this evidence itself does not make NCU stand out from these three organizations. Looking closer we found that 25% of Chen's co-authors belong to MSU, 14% of them belong to NTU while none of them belong to NCU. As to the "citationship", 6% of the papers cited by Chen's paper are from MSU, 2.6% from NTU while, again, none is from NCU. These facts make MSU and NTU closer to each other from our outlier detector's point of view and, thus, NCU stands out. Intuitively this seems reasonable, since we would expect one would have more co-authorships and citation-ships with the people from the same organization. UNICORN discovered that these three organizations are abnormal to the other 960 ones in the sense that Chen only belongs to them. Furthermore, it finds that NCU is "abnormal among the abnormal", since it is different from the other two. UCSB (5th outlier) is also one organization worthy of noticing, since it contributes the 2nd most to the papers that cite Chen's work and the 3rd most to the papers that are cited by his papers. The above facts together with the one that "Chen has never co-authored with any person at that institution" make UCSB a high-ranked outlier.

2.5 Discussion

In the implementation of UNICORN, users have the option of selecting a set of "meta-constraints" to generate the path types used by the system. One such meta-constraint is the maximum length of a path, others allow to constrain the type of nodes used in a path, etc.. Then, given a set of meta-constraints and a particular dataset, the system will automatically generate all path types to be used as features.

Although UNICORN performs "as if" some sort of the semantics of the relations were encoded, some domain knowledge of the bibliography domain was given, and some kind of reasoning system was used within the tool, in fact, none of this is the case. In fact UNICORN is a general, domain independent discovery tool that is not designed specifically for any task. It does not know or have a model of the semantic meaning of paths, or the various inferential capabilities we have used to explain its results. For example, it does not know that if one belongs to an organization, one typically has other connections with that organization, but it performs as if it does know when picking up the abnormal instances. This shows that our method can discover some interesting, deeper logical (even contradictory) relationships, which justifies our approach.

Analyzing the experimental results, we found that in the HEP-Th dataset, UNICORN's discoveries could be further categorized into two subgroups: The first group contains nodes that are *significantly* connected to a source and the second are nodes that are *atypically* connected. In other words, for this dataset the term "abnormal" can be interpreted as either "significant" or "atypical". For example, U. Texas is significantly connected to Pope and so is MSU to Chen. The reason that the nodes contributing significantly are prominent is that in this bibliography dataset people tend to work with a small number of others, they belong to only a few institutions and usually only focus on a specific research topic. Thus the nodes that are significantly connected with the source turn out to be "abnormal". On the other hand, our program also detects atypical nodes such as INFN for Pope, NCU and UCSB for Chen. These nodes do not contribute the most, but they are picked because they contribute atypically. We also found that in most of the cases we can easily verify significant nodes through the Web, but not so for atypical ones. In our opinion this does not mean that the atypical instances discovered are incorrect, on the contrary, they potentially contain interesting information that would be difficult to discover otherwise.

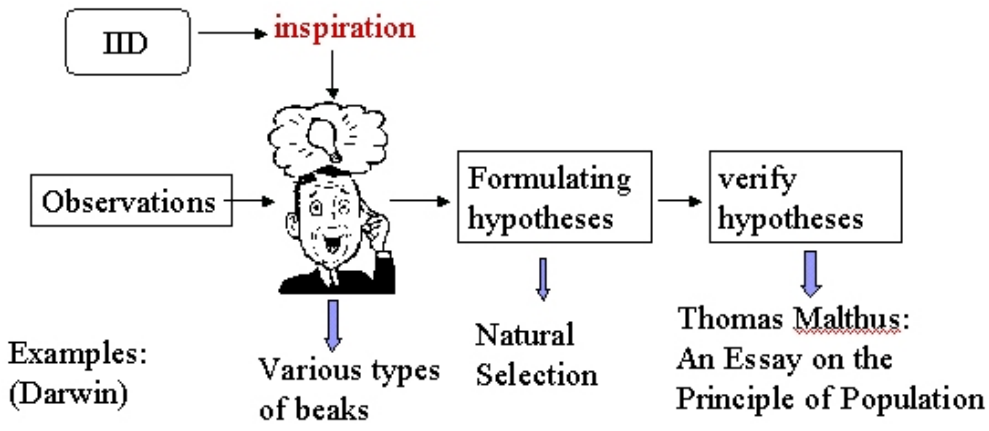


Figure 4: Inspiration-driven discovery

2.6 Applications

Being able to find abnormal, unusual, "suspect" instances is an important capability in a variety of domains such as law enforcement, homeland security, fraud detection and also data cleaning. For example, a murderer might connect to the victim in an abnormal way; a threat event might contribute differently from the normal ones in various aspects; a fraud or intrusion in general behaves differently than normal ones, and so is a typo leading to inconsistency in the data.

Yet it is this more general capability that can support the discovery by humans or machines that make an IID program useful. Figure 4 tries to illustrate that with a cartoon depicting an "inspiration-driven" discovery process for human discovery: the discoverer first has some problems to be solved in mind, and suddenly something interesting (we call it inspiration) occurs to him/her that triggers a specific hypothesis. Afterwards further experiments are performed to verify the hypothesis. History shows that in many cases interesting instances can be an inspiration. For example, the various shapes of beaks (i.e. the interesting instances) Darwin saw on Galapagos triggered his novel thought of evolution theory. Therefore, it is not hard to image that developing a program such as UNICORN to identify interesting instances might be a crucial step towards real machine discovery.

2.7 Related Work

Many science discovery tools such as BACON (Langley et al., 1987) and AM (Lenat, 1982) aim at discovering laws in a specific domain. GRAFFITI (Fajtlowicz, 1988) is a famous discovery program in mathematics. It has successfully generated hundreds of conjectures about inequalities in graph theory by heuristic search, many of which lead to publications when mathematicians tried to prove or refute these conjectures. MECHEM (Vlades-Perez, 1995) is a discovery tool that hypothesizes the structural transformations of chemicals. ARROWSMITH (Smalheiser & Swanson, 1998) is a literature-based discovery tool that hypothesizes possible treatments or causes of diseases using a collection of titles and abstracts from the medical literature. Unlike most other machine discovery approaches, it does focus on instance discovery, but its search criteria are very different from ours, which allows it to use a simpler search method to find associations between treatments and diseases.

Within the area of KDD there is a significant body of work focusing on the discovery of interesting patterns and rules (Freitas, 1999; Silberschatz & Tuzhilin, 1996) as well on the detection of considerably dissimilar or outlier points in data sets (Knorr & Ng, 1998; Ramaswamy et al., 2000). However, the approaches used are not suitable for multi-relational, non-numeric data sets. On the other hand, the concept of "unexpectedness" or "surprise" has been exploited in various discovery problems (Liu, 2001; Silberschatz & Tuzhilin, 1996). In their framework an event is unexpected if it is contrary to the user or analyst's belief which must be known. UNICORN, on the other hand, can find unexpected instances without having to model the user's internal belief. It does so by exploiting "abnormality" to model "unexpectedness" with the intuition that an instance that is very different from its peers has higher possibility to surprise the user. A somewhat similar idea was used by Keogh's work on surprising time series patterns' identification (Keogh, 2002) and Bing Liu's work on discovering unexpected information from websites (Liu, 2001), however, their approach either focuses on pattern discovery or does not handle multi-relational data. Finally, work on link discovery (Mooney et al., 2003), relational and multi-relational data mining (Dzeroski & Lavrac, 2001) as well as social network analysis (Wasserman & Faust, 1994) is also related. However, current

research still focuses on detecting and tracking groups, mining interesting relational association/classification rules or pattern matching.

2.8 Summary

In this section we described the UNICORN program and a new research direction in machine discovery, interesting instances discovery, which aims at finding interesting or suspect instances in large semantic graphs. We claim that it is required to apply an unsupervised method for IID problems since the term “interestingness” is too vague for people to generate any unbiased training examples. UNICORN is an unsupervised instance discovery tool that finds interesting instances in multi-relational data by identifying those with abnormal semantic profiles. UNICORN is able to transform an instance discovery problem in a multi-relational world into a numerical outlier detection problem by summarizing the semantic graph structure surrounding a particular instance. Our method does not require any domain-specific background knowledge or training examples. The case study on a large natural dataset in the bibliography domain shows that our methods can in fact extract instances that are interesting in the real world without actually knowing the semantics of the relations or any background domain knowledge. We also point out that interesting instances can play an important role in an “inspiration driven discovery process” by serving as inspirations during the process of data exploration. Potential applications for our work are in homeland security, law enforcement, fraud detection and data cleaning.

3 KOJAK Group Finder

The KOJAK Group Finder is a hybrid link discovery (LD) system that combines state-of-the-art knowledge representation and reasoning (KR&R) technology with statistical clustering and analysis techniques from the area of data mining. In this section we describe the architecture and technology of the Group Finder in more detail. The Group Finder is capable of finding hidden groups and group members in large evidence databases. Our group finding approach addresses a variety of important LD challenges, such as being able to exploit heterogeneous and structurally rich evidence, handling the connectivity curse, noise and corruption as well as the capability to scale up to very large, realistic data sets.

3.1 Motivation

The development of information technology that could aid law enforcement and intelligence organizations in their efforts to detect and prevent illegal and fraudulent activities as well as threats to national security has become an important topic for research and development. Since the amount of relevant information, tips, data and reports increases daily at a rapid pace, analyzing such data manually to its full potential has become impossible. Hence, new automated techniques are needed to take full advantage of all available information.

One of the central steps in supporting such analysis is link discovery (LD), which is a relatively new form of data mining. Link discovery can be viewed as the process of identifying complex, multi-relational patterns that indicate potentially illegal or threat activities in large amounts of data. More broadly, it also includes looking for not directly explainable connections that may indicate previously unknown but significant relationships such as new groups or capabilities (Senator, 2002) .

Link discovery presents a variety of difficult challenges. First, data ranges from highly unstructured sources such as reports, news stories, etc. to highly structured sources such as traditional relational databases. Unstructured sources need to be preprocessed first

either manually or via natural language extraction methods before they can be used by LD methods. Second, data is complex, multi-relational and contains many mostly irrelevant connections (connectivity curse). Third, data is noisy, incomplete, corrupted and full of unaligned aliases. Finally, relevant data sources are heterogeneous, distributed and can be very high volume.

The KOJAK Group Finder addresses these challenges by combining state-of-the-art knowledge representation and reasoning (KR&R) technology with statistical techniques from the area of data mining. Using KR&R technology allows us to represent extracted evidence at very high fidelity, build and utilize high quality and reusable ontologies and domain theories, have a natural means to represent abstraction and meta-knowledge such as the interestingness of certain relations, and leverage sophisticated reasoning algorithms to uncover implicit semantic connections. Using data or knowledge mining technology allows us to uncover hidden relationships not explicitly represented in the data or findable by logical inference, for example, entities that seem to be strongly related based on statistical properties of their communication patterns.

The Group Finder is capable of finding hidden groups and group members in large evidence databases. Our group finding approach addresses a variety of important LD challenges, such as being able to exploit heterogeneous and structurally rich evidence, handling the connectivity curse, noise and corruption as well as the capability to scale up to very large, realistic data sets. The KOJAK Group Finder has been successfully tested and evaluated on a variety of synthetic datasets with up to 100,000,000 binary links.

3.2 The Group Detection Problem

A major problem in the area of link discovery is the discovery of hidden organizational structure such as groups and their members. There are, of course, many organizations and groups visible and detectable in real world data, but we are usually only interested in detecting certain types of groups such as organized crime rings, terrorist groups, etc.

Group detection can be further broken down into (1) discovering hidden members of *known groups* (or group extension) and (2) identifying completely *unknown groups*.

A known group (e.g., a terrorist group such as the RAF) is identified by a given *name* and a set of known members. The problem then is to discover potential additional hidden members of such a group given evidence of communication events, business transactions, familial relationships, etc. For unknown groups neither *name* nor known members are available. All we know are certain suspicious individuals ("bad guys") in the database and their connection to certain events of interest. The main task here is to identify additional suspicious individuals and cluster them appropriately to hypothesize new real-world groups, e.g., a new money laundering ring. While our techniques address both questions, we believe group extension to be the more common and important problem.

Another important problem characteristic that influenced our solution approach concerns the data. Evidence available to law enforcement organizations is split into *primary* and *secondary* sources. Primary evidence is lower volume, high reliability, usually "owned" by the organization and can be searched and processed in arbitrary ways. Secondary evidence is usually not owned by the organization (e.g., might come from news articles or the Web), is higher volume, might only be searchable in restricted ways and might be associated with a cost (e.g., access might require a warrant). Our group detection approach needs to take these different characteristics into account to keep cost at a minimum and properly handle access restrictions to secondary data sources.

3.3 The KOJAK Group Finder

The KOJAK Group Finder is a hybrid logic-based/statistical LD component designed to solve group detection problems. It can answer the following questions:

- How likely is P a member of group G?
- How likely are P and Q members of the same group?
- How strongly connected are P and Q?

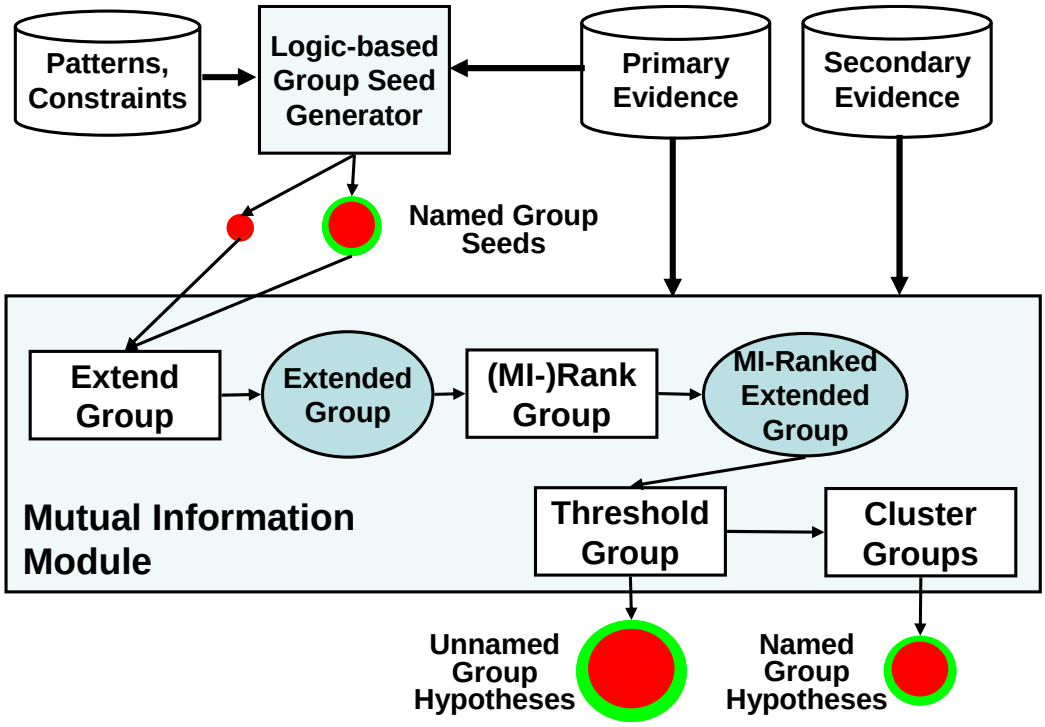


Figure 5: KOJAK Group Finder Architecture

Figure 5 shows the general architecture. The system takes primary and secondary evidence (stored in relational databases) as input and produces group hypotheses (i.e., lists of group members) as output. The system works in four phases. First, a logic-based group seed generator analyzes the primary evidence and outputs a set of seed groups using deductive and abductive reasoning over a set of domain patterns and constraints. Second, an information-theoretic mutual information model finds likely new candidates for each group, producing an extended group. Third, the mutual information model is used to rank these likely members by how strongly connected they are to the seed members. Fourth, the ranked extended group is pruned using a threshold to produce the final output.

The processing for known and unknown groups is somewhat different at the beginning and end of the process. First, the seed generation for unknown groups is different, since there is less information available. Second, the generation of unknown groups involves an

extra step because the extended groups need to be clustered to eliminate duplicates before the thresholding step.

The logic-based seed generation module is based upon the PowerLoom™ knowledge representation & reasoning system (PowerLoom, 2003). The mutual information module was implemented in the STELLA programming language (just like PowerLoom) which can be translated into Common-Lisp, C++ and Java. Evidence databases are stored in MySQL and accessed via JDBC or ODBC.

3.4 The Need for a Hybrid Approach

Link discovery is a very challenging problem. It requires the successful exploitation of complex evidence that comes in many different types, is fragmented, incomplete, uncertain and very large-scale. LD requires reasoning with abstractions, e.g., that **brother-of** and **husband-of** are both subtypes of a **family-relation**, temporal and spatial reasoning, e.g., that cities are subregions of counties which are subregions of states, etc., common-sense type inferences, e.g., that if two people bought tickets for the same event, they probably were at one point in close spatial proximity in the same city, and constrained search, e.g., one might want to look more closely at people who joined a company around the same time a suspect joined. The knowledge and ontologies needed for these types of inferences are very naturally modeled in a symbolic, logic-based approach as done in the logic-based seed generator of the KOJAK Group Finder. However, LD also needs detection and reasoning with statistical phenomena such as communication patterns, behavior similarity, etc., which requires cumulative analysis of evidence that cannot be done in logic but is most effectively done in specialized models such as our mutual information component. Such models, on the other hand, are not well-suited for the representation of complex domains and usually assume some data normalization and simplification. Given these characteristics of the problem, using a hybrid approach that combines the strengths of multiple paradigms is a natural choice. How these two approaches work together for the KOJAK Group Finder is described below

3.5 Logic-Based Seed Generation

The first phase of the KOJAK group detection process is the generation of seed groups. Each seed group is intended to be a good hypothesis for one of the actual groups in the evidence data, even though the number of seed members known or inferable for it might be significantly less than its actual members. The reasons for using this logic-based, seeded approach are threefold. First, the information in primary and secondary evidence is incomplete and fragmented. By "connecting the dots" via logical inference we can extract information that is not explicitly stated and our statistical methods would not be able to infer. Second, because the MI model needs to analyze access-restricted secondary data, it needs good initial focus such as seed groups of "bad guys" in order to query the data selectively. The seeded approach therefore dramatically reduces data access cost as well as MI-processing time. Third, logical reasoning can apply constraints to the information available as well as rule out or merge certain group hypotheses.

To generate seed groups we use the PowerLoom KR&R system to scrub every piece of available membership information from primary evidence (which is smaller volume, less noisy and can be searched arbitrarily). Given the size of primary evidence data we are working with (O(100,000) individuals and O(1,000,000) assertions) we can simply load it directly from the Evidence Data Base (EDB) into PowerLoom using its database interface and a set of import axioms.

The process of finding seeds is different for known and unknown groups. For known groups, we start with a query to retrieve existing groups and their explicitly declared members. We then employ a number of logic rules to infer additional group members by connecting data that is available but disconnected. For example, in the synthetic datasets available to us members of threat groups participate in exploitation cases (meant to model threat events such as a terrorist attack). To find additional members of a group we can look for exploitations performed by a group that have additional participants not explicitly known to be members of the group. The PowerLoom definition below for the

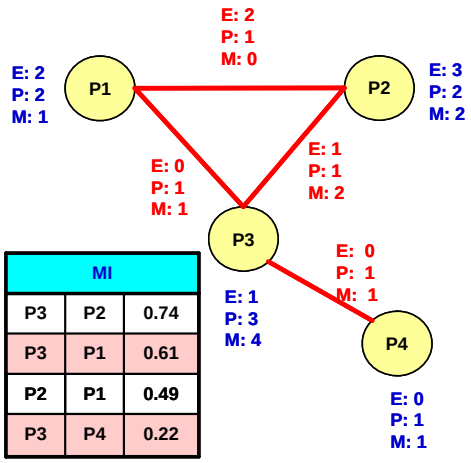
relation **memberAgentsByParticipation** formalizes this type of reasoning (**memberAgents** relates a group and its members; **deliberateActors** relates groups or people to an event):

```
(DEFRELATION memberAgentsByParticipation ((?g Group) (?p Person))
  :<= (AND (Group ?g)
           (Person ?p)
           (FAIL (memberAgents ?g ?p))
           (EXISTS (?c) (AND (ExploitationCase ?c)
                             (deliberateActors ?c ?g)
                             (deliberateActors ?c ?p)))))
```

For unknown groups, we use rules to look for patterns on events to find seeds. The basic idea is to find teams participating in threat events that no (known) group is known to be responsible for. Since people who participate in a threat event are part of a threat group, teams of people who are found to jointly participate in a threat event that cannot be attributed to a known group can be used as seeds for unknown groups. Note, however, that such teams may be subsets of one of the known groups or that two or more of the teams may be part of the same unknown group. For that reason, it is vital to use merging techniques later to combine teams (or their extended groups) if appropriate.

The logic module can also check constraints to help in the merging of hypotheses. For example, a strong hint that two groups may be the same is that their members participated in the same exploitation events. The rule below finds groups who participated in a given exploitation event indicating a potential duplicate group hypothesis if more than one group is found:

```
(DEFRELATION groupHasMemberWhoParticipatedInEvent
             ((?g Group) (?e VulnerabilityExploitationCase))
     :<= (AND (Group ?g)
              (VulnerabilityExploitationCase ?e)
              (EXISTS ?p (AND (Person ?p)
                              (OR (memberAgents ?g ?p)
                                  (memberAgentsByParticipation ?g ?p))
                                  (deliberateActors ?e ?p)))))
```



MI		
P3	P2	0.74
P3	P1	0.61
P2	P1	0.49
P3	P4	0.22

Time	Action	P1	P2	P3	P4
1	P1 Email P2	E	E	Φ	Φ
2	P3 Phone Call P4	Φ	Φ	P	P
3	P2 Meeting P3	Φ	M	M	Φ
4	P1 Email P2	E	E	Φ	Φ
5	P2 Email P3	Φ	E	E	Φ
6	P1 Phone Call P3	P	Φ	P	Φ
7	P1 Phone Call P2	P	P	Φ	Φ
8	P2 Meeting P3	Φ	M	M	Φ
9	P3 Meeting P4	Φ	Φ	M	M
10	P1 Meeting P3	M	Φ	M	Φ
11	P2 Phone Call P3	Φ	P	P	Φ

Figure 6: MI Example. P1, P2, P3 and P4 represent people. E, P and M stand for *Email*, *Phone Call* and *Meeting* respectively. The table on the right shows activities among individuals and the table on the left shows the MI among them.

The use of **memberAgentsByParticipation** shows that these rules not only encode complex queries but also interconnect to build a real domain model. There are about 50 complex rules of this type that are specific to group discovery. Even though the synthetic dataset used in our experiments was designed to be relatively poor in link types and attributes, the data is still quite complex. It contains 72 entity types (22 of which are actually instantiated) and 107 relations and attributes (25 of which are actually instantiated in the data). These entity and relation types are further organized by an ontology (developed by Cycorp) whose upward closure from the entity and relation types in the data contains a hierarchy of about 620 concepts (or classes) and 160 relations. Adding this to the O(1,000,000) assertions representing the evidence we have a fairly large and complex knowledge base to work with.

While the examples given above are specific to the synthetic group discovery domain, the approach is general and applicable to other areas. Evidence data will always be fragmented. Such fragmentation is usually easy to handle by a human analyst, but it can be a big obstacle for an automated system. Using a logic-based model of the domain is a very powerful approach to overcome this problem and connect evidence fragments in useful ways.

3.6 Finding Strong Connections Via a Mutual Information Model

After exploiting the various explicit and implicit evidence fragments given in the EDB to generate a seed group, we try to identify additional members by looking for people that are strongly connected with one or more of the seed members. To find two strongly connected entities, we need to aggregate the many other known links between them and statistically contrast those with connections to other entities or the general population. This cannot be done via a logic-based approach and instead is achieved via an information-theoretic mutual information model.

The mutual information model can identify entities strongly connected to a given entity or a set of entities and provide a ranked list based on connection strength. To do this it exploits data such as individuals sharing the same property (e.g., having the same address) or being involved in the same action (e.g., sending email to each other). Since such information is usually recorded by an observer we refer to it as evidence. Time is often also an important element of evidence and is also recorded in the EDB. Without loss of generality we only focus on individuals' actions in this paper, but not on their properties.

We transform the problem space into a graph in which each node represents an entity (such as a person) and each link between two entities represents the set of actions (e.g., emails, phone calls etc.) they are involved in. For each node we represent the set of its actions with a random variable, which can take values from the set of all possible actions. Figure 6 illustrates this concept. There are four people and three possible actions: sending *Email*, making a *Phone Call* and participating in a *Meeting*. When a person is not involved in any of the above-mentioned actions we indicate that with the empty action φ. For example, we can represent P_1's actions with the random variable X_1 which takes values from the set $\{E, P, M, \varphi\}$ at any given time.

Most individuals in the LD evidence space are connected to each other either directly or indirectly. For example, two people may eat at the same restaurant, drink coffee at the same cafe and take the same train to work every day without any strongly meaningful

connection. On the other hand, three individuals may be strongly connected if they engage in atypical phone call patterns.

To address this problem we measure the *mutual information* (MI) between the random variables representing individuals' activities. MI is a measure of the dependence between two variables. If the two variables are independent, the MI between them is zero. If the two are strongly dependent, e.g., one is a function of another; the MI between them is large. We therefore believe that two individuals' mutual information is a good indicator whether they are in fact strongly connected to each other or not compared to the rest of the population.

There are other interpretations of MI, for example, as the stored information in one variable about another variable or the degree of predictability of the second variable by knowing the first. Clearly, all these interpretations are related to the same notion of dependence and correlation. The *correlation function* is another frequently used quantity to measure dependence. The correlation function is usually measured as a function of distance or time delay between two quantities. It has been shown that MI measures the more general (non-linear) dependence while the correlation function measures linear dependence (Li, 1990). Therefore, MI is the more accurate choice to measure dependence. One of the important characteristics of MI is that it does not need actual variables values to be computed, instead it only depends on the distribution of the two variables. In classical information theory (Shannon, 1948) MI between two random variables *X* and *Y* is defined as:

$$MI(X;Y) = \sum_{x} P(x) \sum_{y} P(y \mid x).\log\left(P(y \mid x) \Big/ \sum_{x} P(x)P(y \mid x) \right)$$

In addition, $MI(X;Y) = H(Y) - H(Y|X) = H(X) - H(X|Y)$, where the conditional entropy $H(X|Y)$ measures the average uncertainty that remains about *X* when *Y* is known (see (Adibi et al. 2004) for more details about the MI model).

3.7 Group Expansion via Mutual Information

Given that we can use the mutual information calculation to find strongly connected individuals, we can exploit this capability to expand the seed groups provided in phase 1 by the logic-based KR&R module. This expansion is done in the following steps:

1 For each seed member in a seed group we retrieve all activities it participates in from primary and secondary data and add any new individuals found to the group. This step therefore expands the seed group graph by one level. Note, that we obey query restrictions for secondary data and only ask one focused query per seed member.

2 Now we view the expanded group as the universe and compute MI for each connected pair in the graph.

3 Next we look for individuals that either have high MI score with one of the seed members or with all seed members when viewed as a single "super individual". Members whose score is below a certain (fairly lax) user-defined threshold are dropped from the list.

4 In this step the MI engine repeats the whole procedure by expanding the expanded group from the previous step one more level and recalculates MI for the new graph. For known groups we stop here and pass the result to the final thresholding step.

5 For unknown groups we usually have much smaller seed sets and therefore repeat the previous step one more time to achieve appropriately-sized group hypotheses.

The group expansion procedure is performed for each seed group generated by the KR&R module and generates an MI-ranked list of possible additional members for each seed group. This list is initially kept fairly inclusive and needs to undergo proper thresholding before it can be reported to a user or passed on to another LD component.

3.8 Threshold Selection and Thresholding

The result of the process described above is a list of extended groups where members are ranked by their mutual information scores. In order to produce and report a definite result on which members we believe are actually part of the group, we need to cut the ordered list at some threshold. The problem is how to set the threshold so that we get

"good" (or even "optimal") recall and precision for a particular application scenario. We used an empirical method that selects a threshold for a dataset based on an empirical analysis of a number of groups in different types of datasets. This method is discussed further in the section describing the experimental results. The good news is that (1) our group detection process generates a very selective ranking (i.e., we reach high recall fairly early) and (2) in real-world situations a good ranking is often more important than picking the best possible cutoff, since human analysts might be willing to accept a certain number of false positives in order to maximize the number of true positives they are after.

3.9 Handling Noise Via a Noisy Channel model

So far we assumed that we are capable to observe all evidence accurately. However, such accuracy occurs rarely in real world databases. We therefore consider the following kinds of noise in the formulation of our model:

Observability (Negative Noise): This factor describes how much of the real data was observable. Not all relevant events that occur in the world will be observed or reported and might therefore not be known to LD components.

Corruption: This type of noise varies from typos to misspelled names all the way to intentional misinformation.

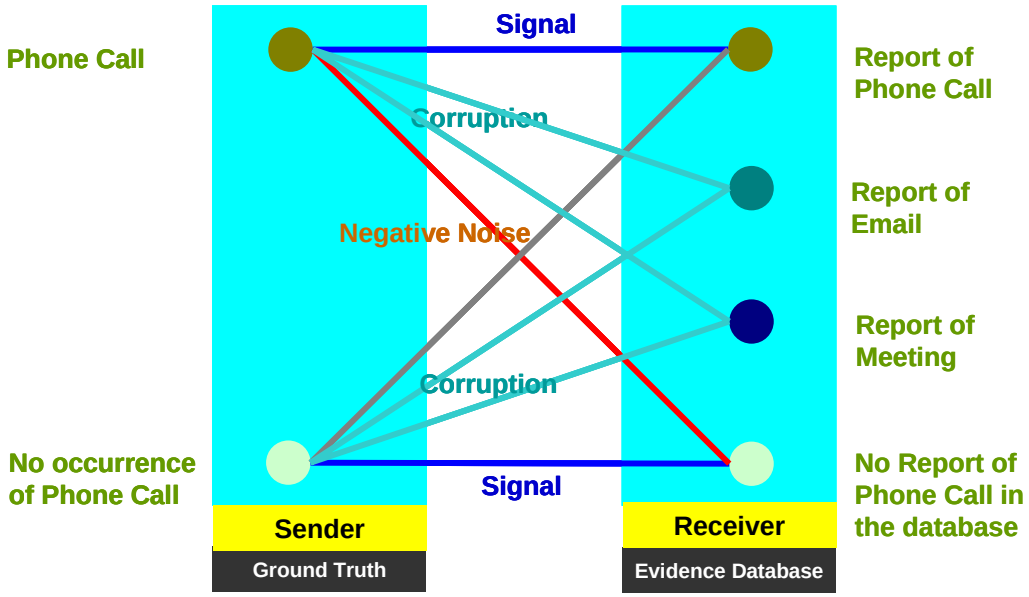


Figure 7: Noise model for a given "Phone Call"

The negative noise phenomenon has been discussed extensively in the communication literature. We adopt the view of a classical noisy channel scenario where a sender transmits a piece of information to a receiver. The transmission goes through a channel with certain noise properties. In our domain we view the ground truth (GT) as the "sender" and the evidence database (EDB) as the "receiver". While in a noiseless environment information is recorded in the EDB without error, in a noisy environment we have a noisy channel, which may alter every piece of evidence transmitted through it with some small probability *p(noise)*. For instance, negative noise occurs if there is a phone call in the ground truth but no record of it in the EDB. Corruption occurs, for example, if there is no phone call in the ground truth but a record indicating one in the EDB. The MI framework is a natural fit for such model. Figure 7 illustrates a noisy channel for a given phone call.

3.10 Complexity and Dataset Scale

Real-world evidence data sets can be very large and we have to make sure that our techniques scale appropriately. The largest synthetic datasets we have analyzed so far contained O(100,000,000) events and O(1,000,000) individuals. Running the KOJAK

GF on such a dataset takes roughly 4 hours on a 2Ghz Pentium-IV desktop with 1Gb of RAM. For more typically sized datasets with 100,000 entities and 1,000,000 links the runtime is in the order of minutes.

The complexity of the MI model is relatively low. The MI engine expands only a limited number of nodes in the problem space starting from the seed members of a group. How many individuals are considered depends on how deeply we grow the link graph to build an extended group. So far, one to two levels have been sufficient. Computing MI between two individuals is $O(N*M)$ where N is the average number of people connected to a given individual and M is the average number of links a person is involved in. Unless N and M grow significantly with larger datasets, the overall complexity is primarily dependent on the number of threat groups we are looking for.

To be able to handle such large datasets in the logic-based seed generation phase, we built a new database access layer into PowerLoom that allows us to easily and transparently map logic relations onto arbitrary database tables and views. By using these facilities we can keep smaller data portions such as the primary data in main memory for fast access and processing, while keeping potentially very large secondary data sets in an RDBMS from where we page in relevant portions on demand. Particular attention was paid to be able to offload large join processing to the RDBMS wherever possible to avoid doing it inefficiently tuple-by-tuple in PowerLoom. This gives us an architecture where we use a traditional RDBMS for storage and access to very large datasets but enrich it with a deductive layer that allows us to formulate more complex queries where necessary. The complexity of the resulting system depends heavily on the nature of the queries and domain rules used which so far has proven to be manageable. For example, the current system uses an ontology with about 800 concept and relation definitions and about 50 complex, non-taxonomic rules that link evidence fragments without any performance problems.

3.11 Experiments

We have applied the KOJAK Group Finder to a wide variety of synthetic data. Access to real world databases has been a main concern in AI, machine learning and data mining communities in the past. The LD community is not an exception in this matter. In particular, since the LD goal is to relate people, place and entities, it triggers privacy concerns. The balance between privacy concerns and the need to explore large volumes of data for LD is a difficult problem. These issues motivate employing synthetic data for performance evaluation of LD techniques.

Synthetic Data

For the purpose of evaluating and validating our techniques, we tested them on synthetic datasets developed by Information Extraction & Transport, Inc. (Silk 2003, Schrag 2003). These synthetic datasets were created by running a simulation of an artificial world. The main focus in designing the world was to produce datasets with large amounts of relationships between agents as opposed to complex domains with a large number of entity properties.

From the point of view of group detection, the artificial world consists of *individuals* that belong to *groups*. Groups can be *threat groups* (that cause *threat events*) or *non-threat-groups.* Targets can be exploited (in threat and non-threat ways) using specific combinations of resources and capabilities; each such combination is called a *mode.* Individuals may have any number of capabilities or resources, belong to any number of groups, and participate in any number of exploitations at the same time. Individuals are *threat* individuals or *non-threat* individuals. Every threat individual belongs to at least one threat group. Non-threat individuals belong only to non-threat groups. Threat groups have only threat individuals as members. Threat individuals can belong to non-threat groups as well. A group will have at least one member qualified for any capability required by any of its modes. Non-threat groups carry out only non-threat modes.

The evidence available in the dataset for our analysis consists of two main types of information:

1 *Individual and group information*. The existence of most individuals and some of the groups is available directly in the evidence. The groups available in the evidence are known or named groups discussed earlier.

2 *Activities from individuals*. Individuals participate in activities related to resources, capabilities and events. Much like in the real world, information about those activities is not available directly, but rather indirectly as transactions (e.g., phone calls or email messages).

Number of entities	**10,000**
Number of Links	**100,000**
Number of Distinct Threat Pattern	**20**
Lowest Signal to clutter ratio	**0.3(-5 db)**
Lowest Signal to Noise Ratio	**.008(-21 db)**
Observability	**50%-100%**
Corruption of Evidence	**0-25%**

Table 2: Synthetic Data Characteristics

Synthetic Data Characteristics

One of the key advantages of using a simulated world is that we are able to test our system against a wide range of datasets. In other words, we are able to create datasets with (almost) arbitrary characteristics, and therefore better understand the potential and limitations of our techniques.

Some of the features used in defining the datasets are in Table 2. The values displayed are typical for the datasets we used in our evaluation; each dataset employs different values for each of these features. Of particular interest are observability (how much of the artificial world information is available as evidence), corruption (how much of the evidence is changed before being reported) and clutter (how much irrelevant information that is similar to the information being sought is added to the evidence).

Evaluation Metrics

The quality of groups we find can be measured with traditional precision and recall metrics defined as follows: Given a proposed group *G* with *g* members which matches an answer group *A* with *a* members, and given that of the *g* proposed members only *c* are correct, precision $P=c/g$ and recall $R=c/a$. Another metric that helps us analyze precision and recall in aggregate is the *F*-measure:

$$F = -\frac{(b^2+1)PR}{b^2P+R}$$

The *F*-measure both requires and allows us to specify the desired trade-off between precision and recall through the *b* variable. A value of $b=1$ indicates that precision and recall are equally important; $b = 2$ means that recall is twice as important as precision, etc. That is, using the *F*-measure allows users of our module to specify their own desired trade-offs in terms of *b*.

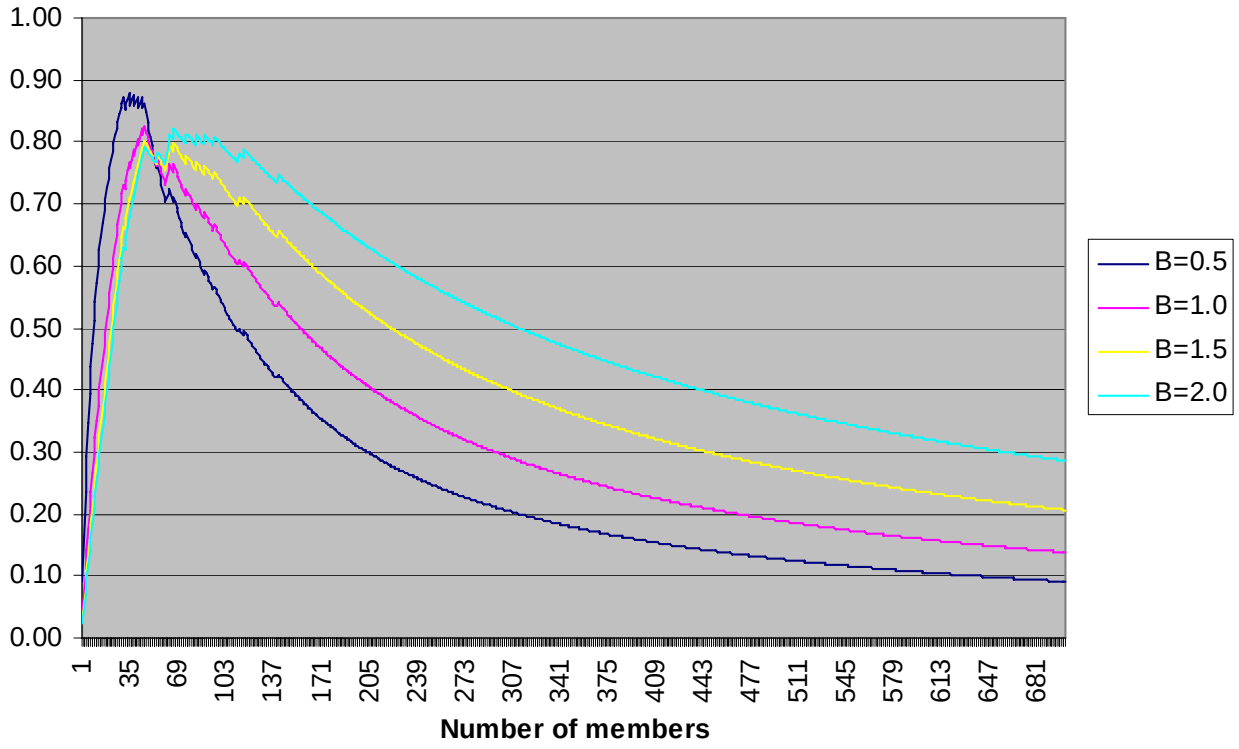


Figure 8: *F*-measure curves for different thresholds for a typical group.

Figure 8 shows a typical set of *F*-measure curves for different thresholds. An important property is that our *F*-measure curves have maximums (and thus optimums). Notice also that *F*-measure curves for higher values of *b* have wider "peaks", which means they are more "forgiving" in threshold selection (a given variation of threshold provokes a smaller variation in F-measure.)

Threshold Analysis

Focusing on the *F*-measure, we defined an empirical model that allowed us to predict good threshold values for a given type of dataset. Datasets vary in many dimensions, in particular on their levels of observability, corruption, and clutter. Our goal was to define a model parametric on these dataset dimensions.

One key initial step is to define the base for the model. Possible bases include the average size of the groups we are looking for (if sufficiently known), the size of extended group and the size of the seed group. Our empirical analysis indicated that the best alternative is to use the size of extended group as a basis for defining the threshold. We

found that the ratio between the real size of the group we would be looking for and the size of the extended group we created as a hypothesis varies little and is usually around 11%-14%. Another advantage is that this measure is organic to the mutual information model, that is, no additional information is needed.

The empirical model consists of defining one specific threshold (as a percentage of the extended group size) for each type of dataset. We used thirteen types of datasets that employed combinations of different values for the parameters in Table 2. We then analyzed the *F*-measure curves to find optimums for each *b*-value (i.e., trade-off between precision and recall) and type of dataset. For example, for a *b* of 1, we predicted a threshold of 8% for a baseline dataset, 6% for a dataset with more clutter, 9% for a dataset with low observability and 3% for a dataset with both additional clutter and low observability. These thresholds are then used to predict the best threshold for a new dataset of a particular type.

Results

We have applied KOJAK to a large number of synthetic datasets of varying complexity and characteristics. Table 3 shows some sample metrics for four datasets. Since there are many groups in each dataset we provide mean and variance values for precision and recall among all groups in a dataset. The average *F*-measure for known groups varies between 0.71 and 0.85. Note that the differences in the properties of the datasets cause the best *F*-measure to be obtained with different recall and precision values. This shows that "harder" datasets, where precision drops more steeply require lower thresholds that yield lower recalls and higher precision values. A more detailed analysis with ROC curves is presented in (Adibi et al. 2004).

		Logic Module		KOJAK Group Finder				
Data set	Number of Groups	Avg. Precision	Avg. Recall	Avg. Precision	Precision Variance	Avg. Recall	Recall Variance	Avg. F-Measure (b=1.5)
Plain	14	1	0.53	0.81	0.005	0.87	0.010	0.85
High clutter	11	1	0.53	0.59	0.010	0.86	0.014	0.74
Low observability	16	1	0.52	0.70	0.004	0.72	0.026	0.71
Both	19	1	0.50	0.88	0.005	0.66	0.011	0.75

Table 3: Scores for applying the KOJAK Group Finder to datasets of increasing complexity (known groups only).

Table 3 also compares the KOJAK results against a baseline of using only the logic module. The results show that the logic module is very accurate (precision = 1), meaning all members found are provably correct. However, since the evidence is incomplete the logic module achieves a maximum recall of about 50%.

We also evaluated our threshold prediction model. We found that the average *F*-measure for these datasets compares to the optimum *F*-measure obtained by using the best possible threshold for each group would result only in a difference of around 6%. In other words, the threshold model only "misses" 6% of whatever was available in the extended groups.

3.12 Related Work

Link discovery (LD) can be distinguished from other techniques that attempt to infer the structure of data, such as classification and outlier detection. Classification and clustering approaches such as that of Getoor et al. (2001) try to maximize individual similarity within classes and minimize individual similarity between classes. In contrast, LD focuses on detecting groups with strongly connected entities that are not necessarily similar. Outlier detection methods attempt to find abnormal individuals. LD, on the other hand, identifies important individuals based on networks of relationships. Additionally, outlier techniques require large amounts of data including normal and abnormal cases, and positive and negative noise. This is inappropriate for LD applications that need to detect threats with few or no available prior cases. Mutual

information has also been used in other domains such as finding functional genomic clusters in RNA expression data and measuring the agreement of object models for image processing (Butte, 2000).

Our work can be distinguished from other group detection approaches such as Gibson, (1998) and Ng, (2001) by three major characteristics. First, our method is unique, since it is based on a hybrid model of semantic KR&R and statistical inference. There are very few approaches that use semantic information. Second, in our approach each type of relation (link) is valuable and treated differently, in contrast to work in fields such as Web analysis and social networks. Third, with our technique, multiple paths between individuals or groups (direct or indirect) imply a strong connection which is different from techniques which focus on finding chains of entities.

The work closest to our own is that of Jeremy Kubica et al. (Kubica, 2002; Kubica, 2003) that uses a probabilistic model of link generation based on group membership. The parameters of the model are learned via a maximum likelihood search that finds a Gannt Chart that best explains the observed evolution of group membership. The approach has a strong probabilistic foundation that makes it robust in the face of very low signal-to-noise ratios.

Another recent approach to the LD problem is the use of probabilistic models (Cohn, 2001; Friedman, 1999; Getoor, 2001). Kubica et al. (2001) present a model of link generation where links are generated from a single underlying group and then have noise added. These models differ significantly from ours since we do not assume a generative model of group formation, but rather probabilistically determine each entity's membership.

3.13 Summary

In this section we described the KOJAK Group Finder (GF) as a hybrid model of logic-based and statistical reasoning. GF is capable of finding potential groups and group

members in large evidence data sets. It uses a logic-based model to generate group seeds and a multi-relational mutual information model to compute link strength between individuals and group seeds. Noise and corruption are handled via a noisy channel model. Our GF framework is scalable and robust, and exhibits graceful degradation in the presence of increased data access cost and decreased relational information.

The Group Finder is best-suited for problems where some initial information or group structure is available (e.g. finding hidden members of existing groups vs. detecting completely new groups) which is a common case in many real world applications. Group detection is useful for law enforcement, fraud detection, homeland security, business intelligence as well as analysis of social groups such as Web communities.

4 References

J. Adibi, H. Chalupsky, E. Melz & A. Valente (2004). The KOJAK Group Finder: Connecting the Dots via Integrated Knowledge-Based and Statistical Reasoning. IAAI 2004.

J. Adibi, H. Chalupsky (2005), Scalable Group Detection via a Mutual Information Model, IA 2005.

Butte, A. & Kohane, I. (2000). Mutual information relevance networks: Functional genomic clustering using pairwise entropy measurements. *Pacific Symposium on Biocomputing*. Honolulu, Hawaii.

Cohn, D. & Hofmann, T. (2001). The missing link: a probabilistic model of document content and hypertext connectivity. *Advances in Neural Information Processing Systems* 13: 430–436.

Dzeroski, S. and Lavrac, N. (2001). *Relational Data Mining*. Berlin: Springer-Verlag.

Fajtlowicz, S. (1988). *On conjectures of Graffiti*. Discrete Mathematics, **72**: p. 113-118.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). *The KDD Process for Extracting Useful Knowledge from Volumes of Data.* Communications of the ACM, **39**(11): p. 27-34.

Freitas, A. (1999) *On rule interestingness measures.* Knowledge-Based System.

Friedman, N., Getoor, L., Koller, D. & Pfeffer, A. (1999). Learning probabilistic relational models. *IJCAI 1999,* San Francisco, Morgan Kaufmann Publishers.

Getoor, L., Segal, E., Taskar, B., & Koller, D. (2001). Probabilistic models of text and link structure for hypertext classification. *IJCAI 2001 Workshop on Text Learning: Beyond Supervision*. Seattle, Washington.

Gibson, D., Kleinberg, J. & Raghavan, P. (1998). Inferring Web communities from link topology. In *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia*. New York, ACM Press.

Hill, R. (1995). Intelligent Systems: Third Golden West International Conference: Edited and Selected Papers, pages. Non-well-founded set theory and the circular semantics of semantic networks.

Keogh, E. J., Lonardi, S., and Chiu, B. (2002). Finding surprising patterns in a time series database in linear time and space. in Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Knorr, E. and Ng, R. (1998). Algorithms for Mining Distance-Based Outliers in Large Datasets. in Proceedings of VLDB.

Kubica, J., Moore, A., Cohn, D. & Schneider, J. (2003). Finding underlying connections: a fast method for link analysis and collaboration queries. *International Conference on Machine Learning (ICML)*.

Kubica, J., Moore, A., Schneider, J. & Yang, Y. (2002). Stochastic link and group detection. *Eighteenth National Conference on Artificial Intelligence (AAAI).*

Langley, P., Simon, H. A., Bradshaw, G. L., and Zytkow, J. M. (1987). *Scientific Discovery: Computational Explorations of the Creative Processes.* Cambridge, MA: MIT Press.

Lenat, D. (1982). *The Nature of Heuristics.* Artificial Intelligence, **19**: p. 189-249.

Li, W. (1990). Mutual information functions versus correlation functions. *Journal of Statistical Physics* 60: 823-837.

Lin, S. & Chalupsky, H. (2003). Using unsupervised link discovery methods to find interesting facts and connections in a bibliography dataset. *SIGKDD Explorations*, 5(2): 173-178.

Liu, B., Ma, Y., and Yu, P. (2001). Discovering Unexpected Information from Your Competitors' Web Sites.

Milosavljevic, A. (1995) *Comments on Herbert Simon's Paper*. Foundations of Science, **2**: p. 201-224.

Mooney, R., Melville, P., Tang, R., Shavlik, J., Dutra, I. d., Page, D., and Costa, V. S., Relational Data Mining with Inductive Logic Programming for Link Discovery, in Data Mining: Next Generation Challenges and Future Directions, H.K.A. Joshi, Editor. 2003, AAAI/MIT Press.

Ng, A., Zheng, A. & Jordan, M. (2001). Link analysis, eigenvectors and stability. *IJCAI 2001*.

PowerLoom (2003). www.isi.edu/isd/LOOM/PowerLoom.

Ramaswamy, S., Rastogi, R., and Kyuseok, S. (2000). Efficient Algorithms for Mining Outliers from Large Data Sets. in Proceedings of the ACM SIGMOD Conference.

Schrag, R. et. al. (2003). EELD Y2 LD-PL Performance Evaluation, Information Extraction and Transport, Inc.

Senator, T. (2002). Evidence Extraction and Link Discovery, *DARPA Tech 2002*.

Shannon, C. (1948). A Mathematical Theory of Communication. *Bell System Tech. Journal* 27: 379-423.

Silberschatz, A. and Tuzhilin, A. (1996). *What Makes Patterns Interesting in Knowledge Discovery Systems.* IEEE Transactions of Knowledge and Data Engineering, **8**.

Silk, B. & Bergert, B. (2003). EELD Evidence Database Description, Information Extraction and Transport, Inc.

Simon, H. (1995). *Machine Discovery*. Foundations of Science, **2**: p. 171-200.

Smalheiser, N. and Swanson, D. (1998). *Using Arrowsmith: a computer-assisted approach to formulating and assessing scientific hypotheses.* Computer Methods and Programs in Biomedicine, **57**: p. 149-153.

Vlades-Perez, R. (1995). *Machine Discovery in Chemistry: New results.* Artificial Intelligence, **74**: p. 191-201.

Wasserman, S. and Faust, K. (1994). *Social Network Analysis: Methods & Applications*., Cambridge, UK: Cambridge University Press.